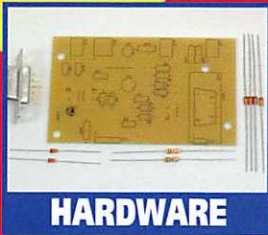
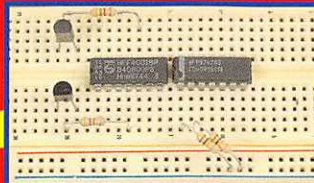
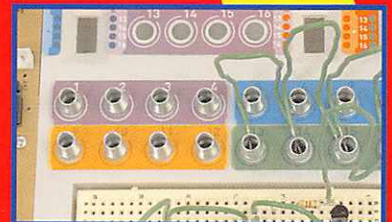
impara l'elettronica digitale

...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

26

9 771824 363008 41026

Peruzzo & C.

Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

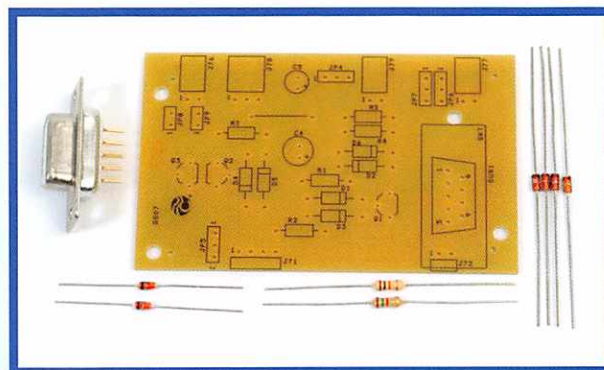
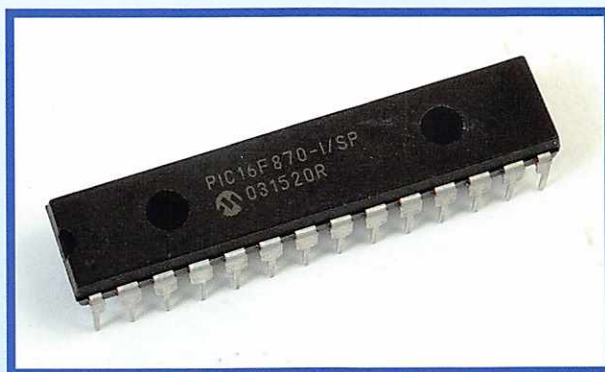
Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

impara l'elettronica digitale

IN REGALO in questo fascicolo

- 1 Scheda DG07
- 1 Connettore femmina SUB-D diritto da 9 pin
- 4 Diodi 1N4148
- 1 Diodo zener 8V2
- 1 Diodo zener 5V1
- 1 Resistenza 1K5 5% 1/4 W
- 2 Resistenze 10K 5% 1/4 W

IN REGALO nel prossimo fascicolo

1 PIC 16F870 programmato

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

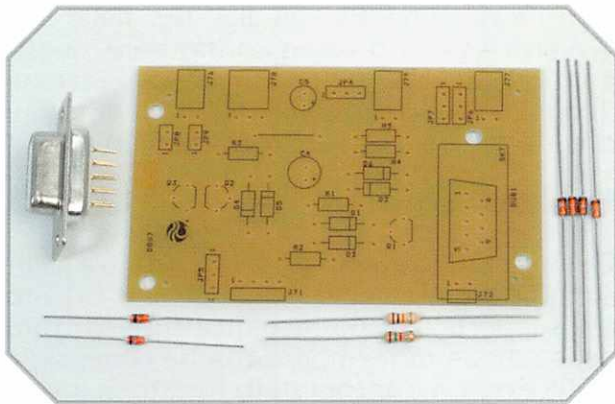
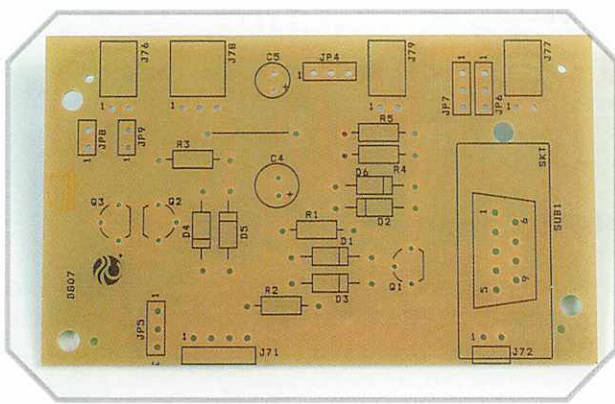
L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede. **Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:** elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Circuito di scrittura

Circuito stampato DG07 e alcuni dei suoi componenti.

Lato dei componenti della scheda DG07.

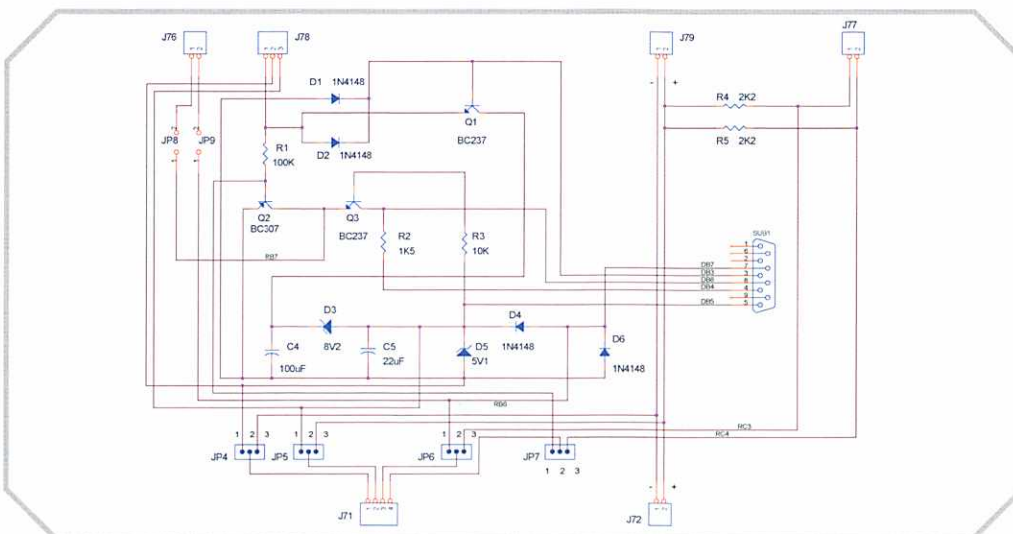
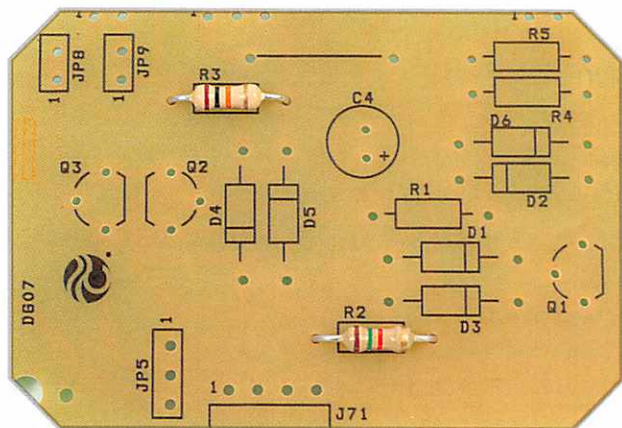
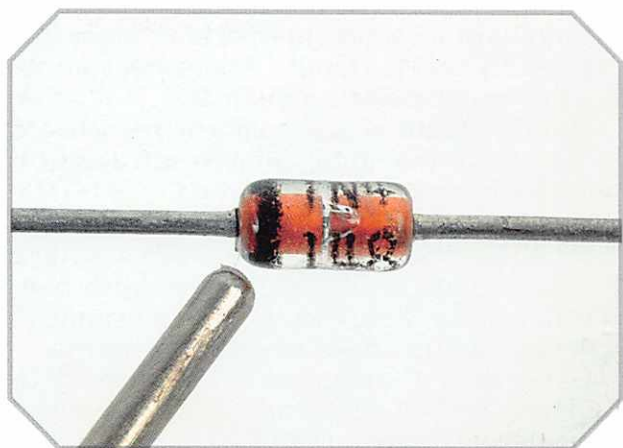
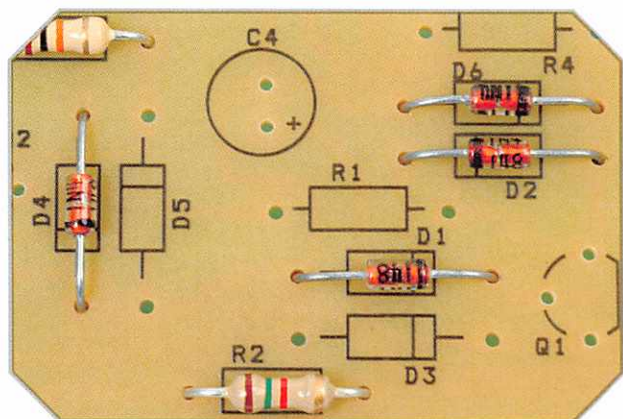
Con questo fascicolo vengono forniti la scheda del circuito stampato DG07, corrispondente al circuito di scrittura del PIC 16F870, e alcuni dei componenti da montare su di essa.

Funzioni

La funzione principale di questo circuito è la scrittura del PIC 16F870, e a questo scopo sarà necessario collegarlo alla porta seriale di un personal computer. Questo circuito fornisce tutte le tensioni necessarie per realizzare la scrittura, con la particolarità di alimentarsi direttamente tramite questa porta, quindi non utilizza nessuna delle alimentazioni del laboratorio. Il programma permette di leggere il contenuto del PIC tramite il personal computer, e di visualizzarlo sul monitor.

Questo circuito si utilizza anche per scrivere la scheda di memoria, sempre attraverso il personal computer.

Inoltre trasferisce l'alimentazione del laboratorio, da 5 V, dalla scheda DG08 fino alla DG06, e semplifica il collegamento tra quest'ultima e la prima, che conterrà il sistema di scrittura e lettura della scheda di memoria.

Le funzioni di scrittura e di funzionamento normale del PIC si configurano mediante i ponticelli della scheda DG06 e quelli della presente scheda, DG07; su quest'ultima sono

Schema elettrico del circuito DG07.

I primi componenti da montare sono le resistenze R2 e R3.

Identificazione del catodo di un diodo 1N4148.

I quattro diodi 1N4148 montati sulla scheda.

identificati come JP4, JP5, JP6, JP7, JP8 e JP9. Vi verrà spiegato come configurare questi ponticelli di volta in volta nella realizzazione di ogni lavoro o esperimento, in modo che il circuito funzioni sempre correttamente.

Posizione

Questo circuito stampato si inserisce tra il circuito principale del PIC, il DG06, e il circuito che supporta il dispositivo di lettura/scrittura delle schede di memoria, che ha come sigla DG08 e non vi è ancora stato fornito. Si installerà in modo definitivo quando vi verranno forniti tutti i suoi componenti. Si fissa con quattro viti sui riferimenti X2, X3, X6 e X7 della zona 2 del laboratorio.

Montaggio

Il montaggio dei componenti su questo circuito stampato si realizzerà man mano che questi vi verranno forniti; vista la quantità, ci vorranno diversi fascicoli.

Resistenze

I primi componenti che monteremo sono le due resistenze fornite: quella da 1K5 (marrone, verde, rosso), nella posizione R2, e quella da 10 K (marrone, nero, arancio) nella posizione R3. Una volta saldati i suoi terminali, taglieremo la parte in eccesso dei reofori.

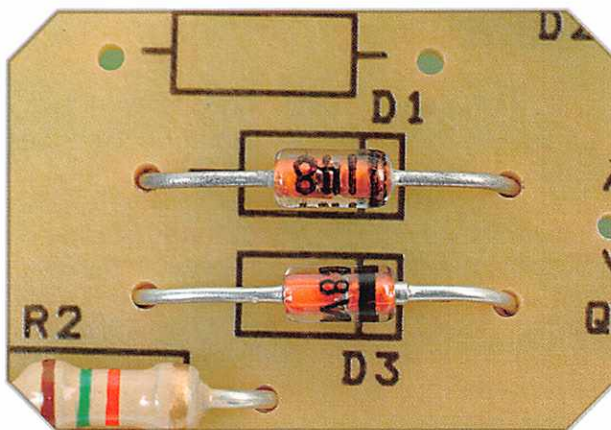
Diodi

Questo circuito stampato contiene 6 diodi, ognuno dei quali, con le dovute precauzioni, verrà montato al suo posto e con l'orientamento corretto.

I diodi D1, D2, D4 e D6 sono uguali fra loro e del tipo 1N4148, il catodo del diodo si distingue perché è il terminale più vicino all'anello nero stampato sul contenitore dello stesso. Nella serigrafia della scheda questa banda è indicata da un tratto, il quale attraversa il rettangolo che rappresenta il diodo.

Diodi zener

I diodi D3 e D5 sono diodi zener, e a loro volta sono di due tensioni differenti, D3 è da 8,2 V, mentre D5 è da 5,1 V. D3 si identifica dalla sigla 8V2 stampata sul suo contenitore, men-

Diodo zener 8V2.

Diodo zener 8V2 installato sulla scheda.

tre D5 ha la sigla 5V1; in alcuni modelli può essere necessario utilizzare una lente di ingrandimento per leggere queste serigrafie d'identificazione.

Il connettore

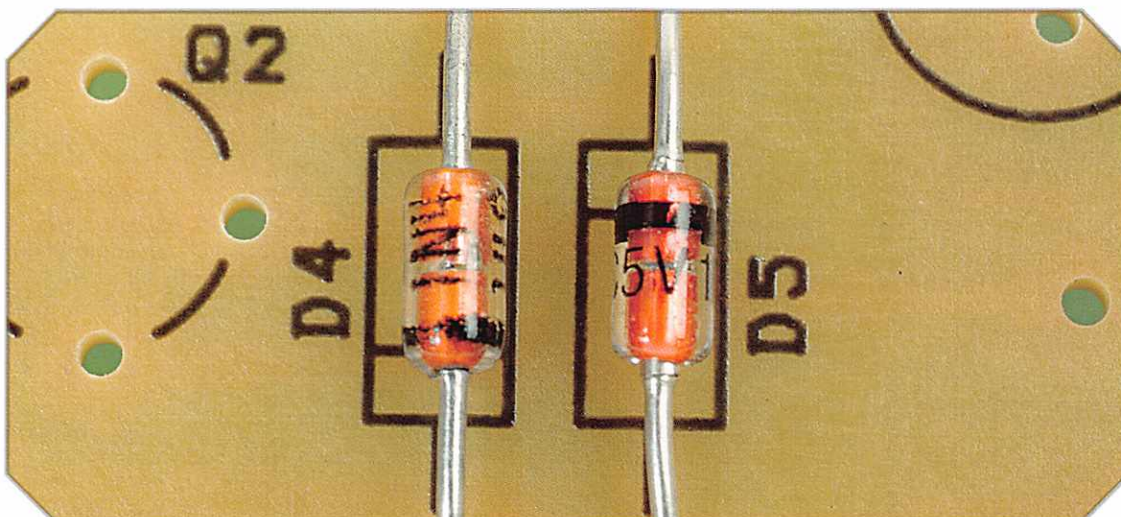
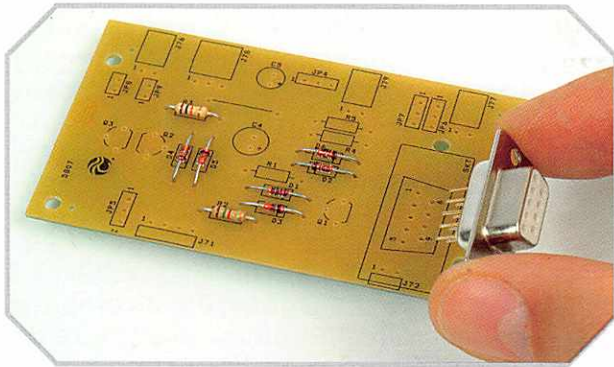
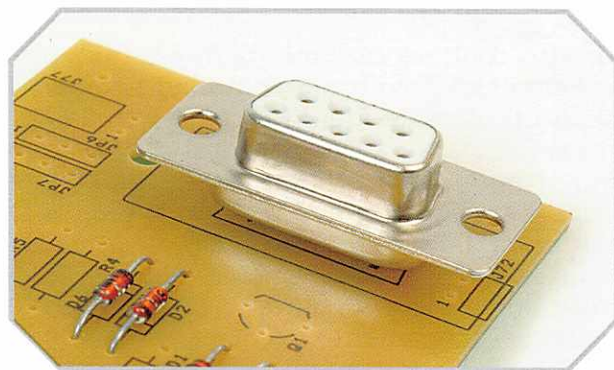
Questa scheda contiene un connettore femmina tipo Sub D a 9 pin con cui si realizza il collegamento tra il PIC 16F870 e il computer dove è installato il programma di scrittura del PIC. Il collegamento si realizza con un cavo standard di connessione pin a pin terminato su due connettori maschio.

Installazione del connettore

Questo connettore si monta inserendo i suoi terminali sul circuito stampato nella zona che si può facilmente identificare grazie alla forma serigrafata del connettore stesso. È molto importante che il connettore sia totalmente appoggiato alla scheda quando si esegue la saldatura dei suoi terminali, allo scopo di fissarlo in modo sicuro ed evitare movimenti che potrebbero piegarlo durante qualche operazione di collegamento e scollegamento.

Collegamento a DG06

Questo circuito stampato si collega alla scheda DG06 tramite i connettori J76, J78, J79 e J77, che collegano direttamente i connettori J66, J68, J69 e J67 di DG06.

Il connettore J76 si utilizza per l'ingresso

Diodo zener da 5V1 installato.

Installazione del connettore.

Il connettore deve essere appoggiato alla scheda.

dei dati di scrittura e si collega, configurando i ponticelli in modo adeguato, alle porte dei PIC RB6 e RB7.

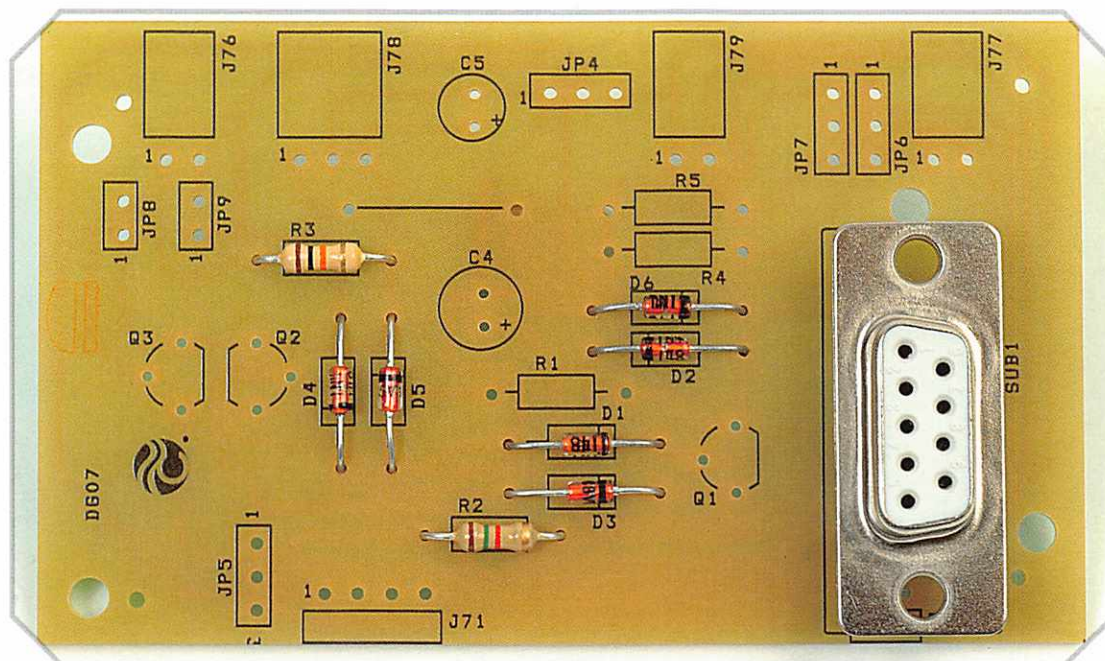
Il connettore J78 è di alimentazione per la scrittura, e permette di far arrivare fino al PIC le tensioni necessarie per il processo di scrittura, generate dal circuito di scrittura che stiamo costruendo sulla scheda DG07.

Il connettore J79 si utilizza per trasferire l'alimentazione del laboratorio, dalla scheda DG08 fino alla DG06; questa è l'alimentazione che viene utilizzata normalmente, fatta eccezione per i processi di scrittura, in cui si utilizza quella del computer.

Il connettore J77 si utilizza per il trasferimento dei dati tra la scheda di memoria e le porte RC3 e RC4 del PIC.

Connessione a DG08

Il collegamento alla scheda DG08 si realizza con soli due connettori, uno di essi, il J71 a quattro vie, si utilizza per il collegamento alla scheda di scrittura e lettura della scheda di memoria.

Il connettore J72 prende l'alimentazione da 5 V dalla scheda DG08, e quest'ultima, a sua volta, la ottiene dall'alimentazione a 5 V del laboratorio. Il terminale 1 di J72 corrisponde a 0 V dell'alimentazione, e il terminale 2 di J72 al positivo di 5 V dell'alimentazione.

Condizione attuale della scheda in attesa di ulteriori componenti.

Semisommatore

In questo esercizio faremo delle prove con un circuito matematico, il semisommatore. Questo circuito si utilizza per ottenere la funzione somma, inoltre indica se c'è o meno il riporto.

Il semisommatore

Questo circuito dispone di due terminali di uscita, uno per la somma e l'altro per il riporto. Non ha ingresso di riporto per somme precedenti, quindi non è un sommatore completo o "full adder", è solamente un semisommatore, ovvero un "half adder".

Il circuito

Seguendo lo schema del circuito si può ottenere la tabella della verità dello stesso, la funzione somma si ottiene sull'uscita della porta U1B, che corrisponde al terminale 4 del circuito integrato U1.

Gli ingressi sono rappresentati dai contatti A e B, quando questi contatti sono aperti abbiamo 0 sull'ingresso, in quanto il livello basso è mantenuto dalle resistenze di "pull down"; quando i contatti sono chiusi, invece, sugli ingressi abbiamo il valore 1.

Sull'uscita della porta U1C si ottiene il bit di riporto della somma, che è 1 quando i due ingressi sono a 1.

Bisogna evitare di confondere questa funzione matematica con la funzione logica OR, anche se sono molto simili.

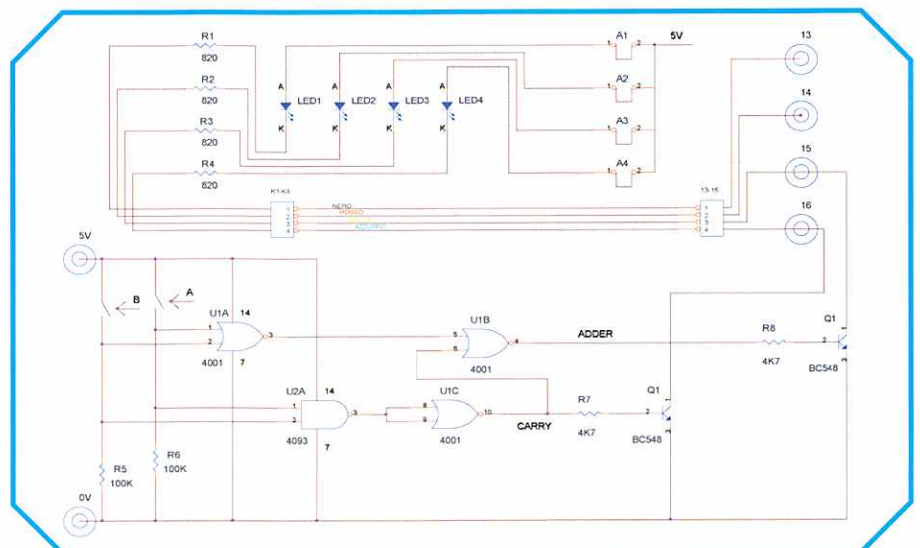
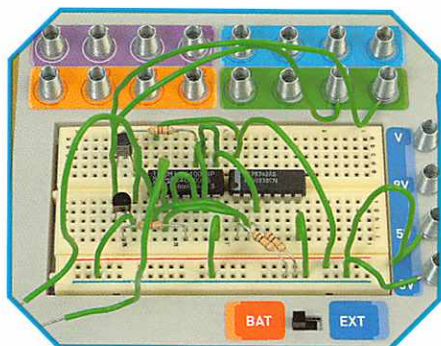
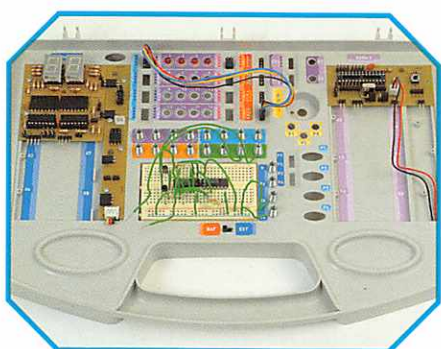
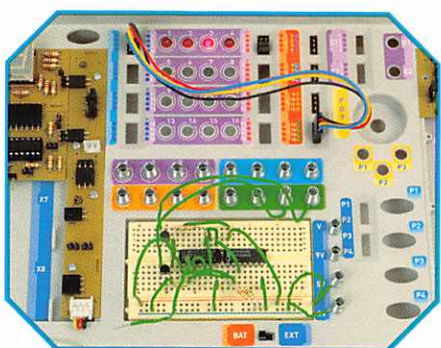
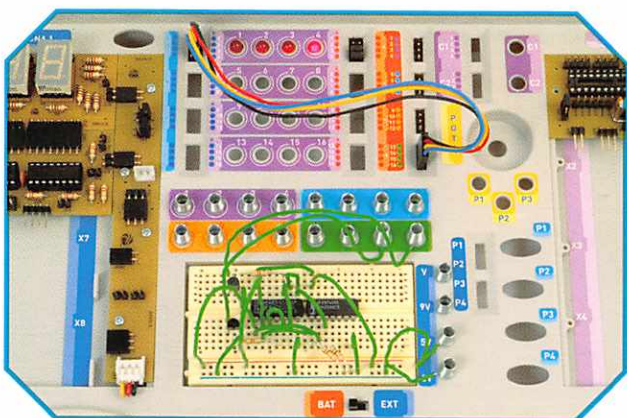
La differenza consiste nel fatto che quando i due ingressi sono a 1 l'uscita della porta OR è 1, mentre nel sommatore l'uscita è 0 e il bit di riporto è 1, che andrebbe a sommarsi al sommatore del bit successivo, quindi dovrebbe necessariamente essere un sommatore completo.

Fili di collegamento installati.

A	B	SOMMA (ADDER)	RIPORTO (CARRY)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Funzionamento

Il funzionamento del circuito segue la tabella della verità del semisommatore. Ricordate che ci sono due uscite, e per poter visualizzare l'uscita del risultato della somma (adder) si utiliz-

Schema elettrico del semisommatore.

I collegamenti A e B sono gli ingressi del circuito.

L'alimentazione si prende dalle molle 0 V e 5 V.

Collegamento fisso di A con B scollegato, la somma è 1 e il LED 3 si illumina.

Per fare in modo che ci sia riporto i due ingressi devono essere a 1.

za il LED 3, che si illumina quando il risultato della somma è 1. Il LED 4 si utilizza per rilevare quando c'è riporto, ovvero quando il riporto è 1, in questo caso si illumina e questo succede quando i due ingressi sono posti a livello logico 1: quando i contatti A e B sono chiusi.

Montaggio

In questo esperimento si utilizzano due circuiti integrati, che sono i primi componenti da inserire sulla scheda Bread Board. Di seguito inseriremo le resistenze e i transistor, dopodiché si realizzeranno i collegamenti interni della scheda e poi quelli che vanno alle molle 15, 16 e 0 V. Prima di collegare il 5 V è consigliabile rivedere tutto il lavoro svolto per essere certi di non aver commesso alcun errore.

Occorre collegare un cavetto, terminante su due connettori a quattro vie, tra i terminali 13-16 e i terminali dei primi quattro LED della matrice; monteremo anche i ponticelli su A3, A4 e il terminale che hanno a fianco, collegato internamente a 5 V, identificato solamente con il simbolo "+".

Alimentazione

L'ultimo passo consiste nel collegare le molle di alimentazione da 5 V e verificare lo stato delle pile collegate al portabatterie situato sotto la zona 1, prima di passare il commutatore di alimentazione sulla posizione BAT.

Prova

Utilizzare i collegamenti A e B per verificare che si compia la tabella della verità. Ad esempio, con A e B scollegati il risultato della somma è 0 e il bit di riporto è anch'esso a 0, quindi non si deve illuminare nessun LED. Se colleghiamo A e B si illuminerà il LED 4, indicando che il bit di riporto è a 1. Se colleghiamo solo A, o solo B, si illuminerà il LED 3, il quale rappresenta la somma, indicando che è a 1, e il LED 4 rimane spento indicando, che il bit di riporto è a 0.

LISTA DEI COMPONENTI

U1	Circuito integrato 4001
U2	Circuito integrato 4093
Q1, Q2	Transistor BC547 o BC548
R5, R6	Resistenza 100 K (marrone, nero, giallo)
R7, R8	Resistenza 4K7 (giallo, viola, rosso)

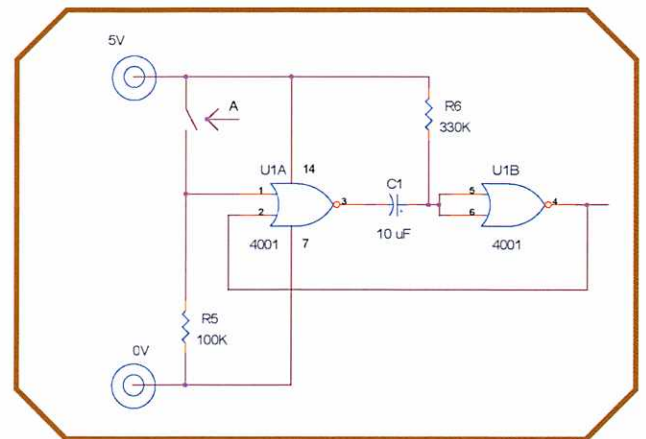
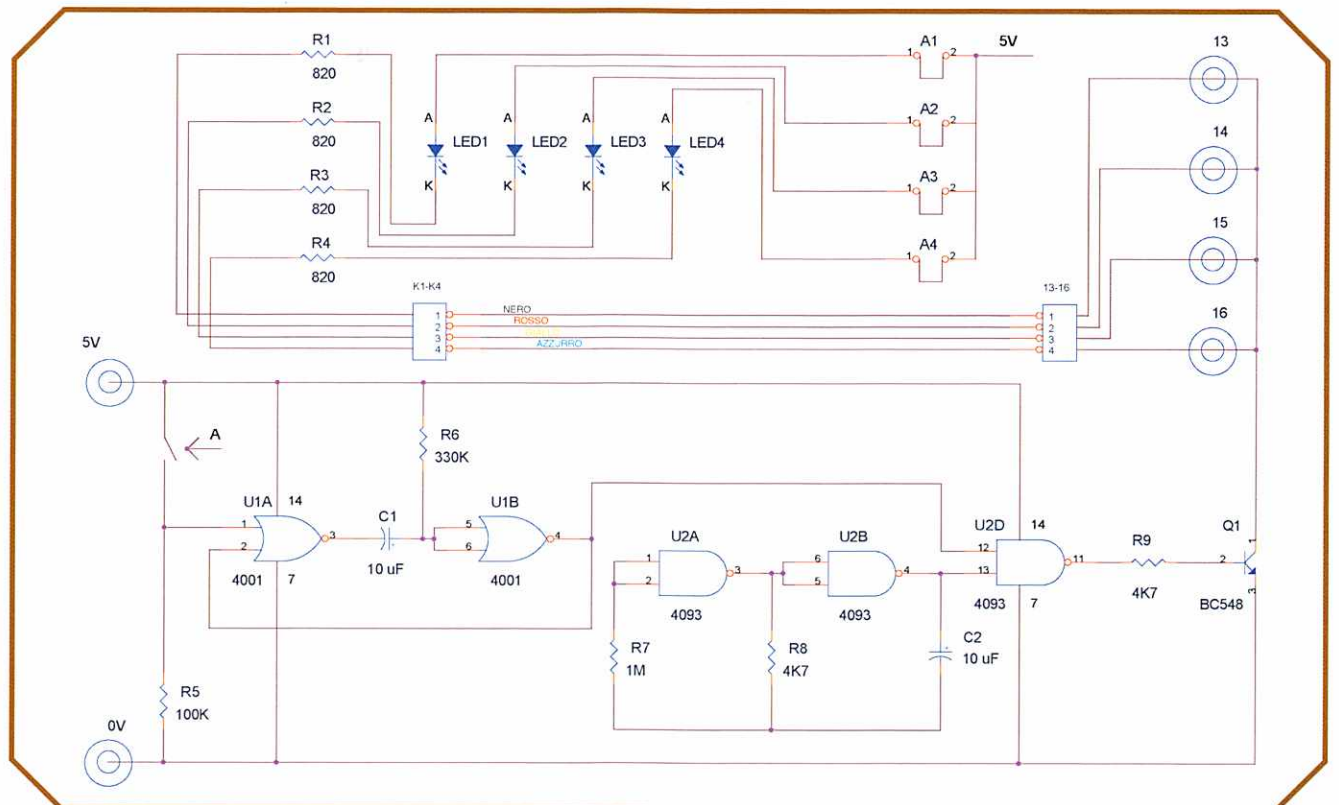
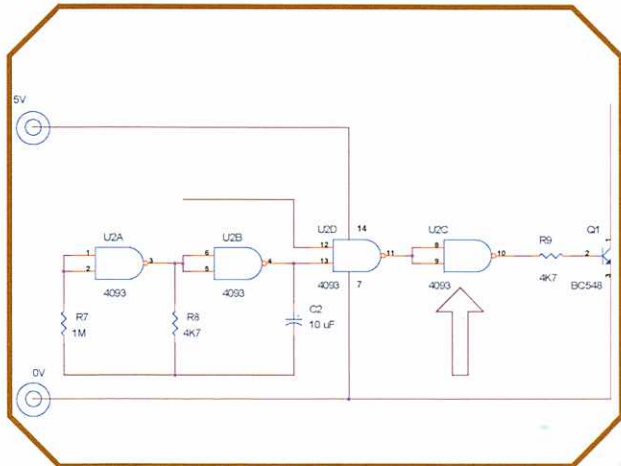
Indicatore temporizzato

Una delle molte applicazioni dei circuiti astabili e monostabili è l'indicazione dello stato di alcuni circuiti, ad esempio gli allarmi; faremo delle prove con un circuito che utilizza entrambi i tipi di multivibratori.

Funzionamento

Il circuito ha un ingresso di attivazione rappresentato dal collegamento A, e quattro LED che rimangono illuminati. Quando si attiva l'ingresso A, collegando il filo per un attimo, i LED si illuminano in modo intermittente per un certo periodo, dopodiché tornano a un'illuminazione fissa fino a quando si attiverà nuovamente l'ingresso del circuito.

Il circuito

Se osserviamo lo schema vediamo che il collegamento A attiva un monostabile formato

Monostabile.

Schema del circuito dell'indicatore temporizzato.

Aggiungendo una porta si inverte l'illuminazione

Scheda Bread Board con i componenti inseriti.

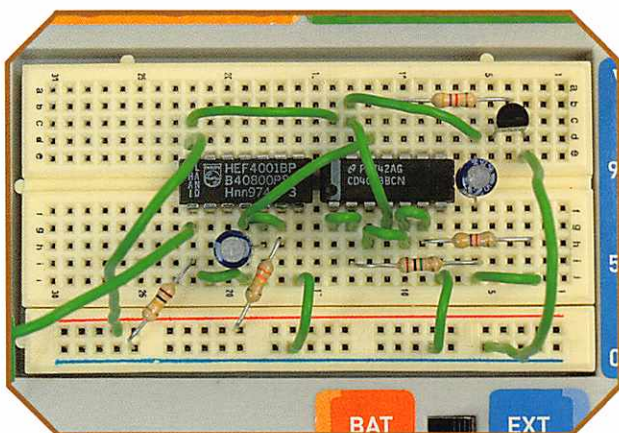
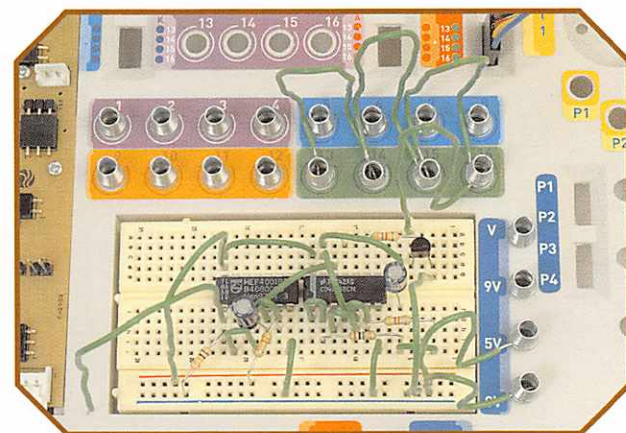
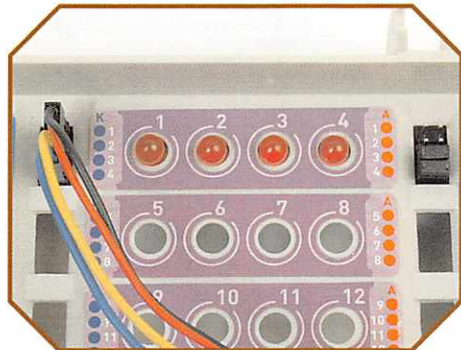
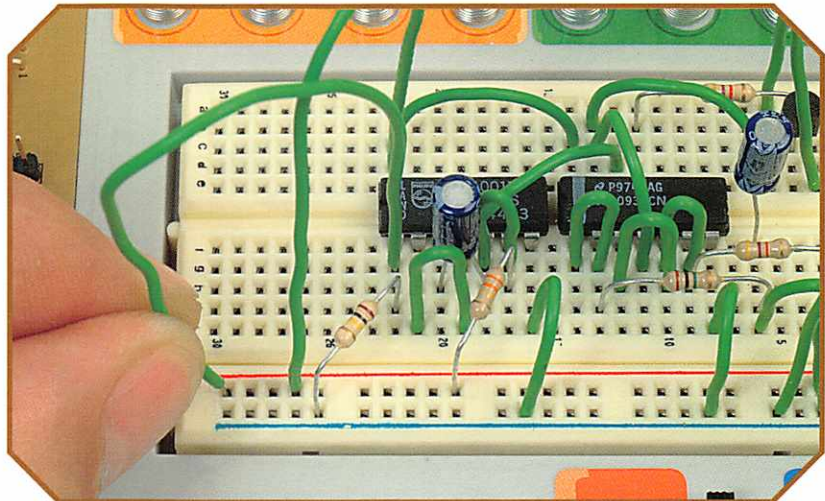
dalle porte U1A, U1B, il condensatore C1 e la resistenza R6. I valori di questi ultimi componenti determinano la durata di questa temporizzazione.

Nel circuito possiamo vedere anche un oscillatore astabile, formato dalle porte U2A e U2B, la cui uscita è portata a uno degli ingressi della porta U2D, e lo stato dell'altro ingresso dipende dall'uscita del temporizzatore: quando l'uscita di questa porta è uno, si polarizza il transistor Q1 in modo da illuminare contemporaneamente i LED da 1 a 4. Gli anodi dei LED sono collegati al positivo dell'alimentazione.

Montaggio

Il montaggio si deve eseguire con molta attenzione per evitare errori. È conveniente iniziare inserendo i circuiti integrati e il transistor, facendo molta attenzione al suo orientamento. Dopodiché installeremo le resistenze e i condensatori elettrolitici, osservando la polarità di questi ultimi: il negativo è indicato sul contenitore degli stessi, inoltre il terminale positivo è più lungo, a meno che non sia stato tagliato per qualche motivo.

L'alimentazione del positivo dei LED si ottiene collocando i quattro ponticelli sui terminali A1, A2, A3 e A4, e i terminali situati a fianco e identificati con il simbolo "+". Questi terminali sono collegati direttamente al positivo dell'alimentazione da 5 V.

È necessario, inoltre, collegare un cavetto a quattro fili con due connettori a quattro vie, sui terminali dal 13 al 16 e sui terminali da K1 a K4; in questo modo il collegamento dei catodi di LED 1 fino a LED 4 si trasferisce alle

Cablaggio interno.

Collegamenti alle molle.

Dettaglio dei ponticelli che portano l'alimentazione agli anodi dei LED.

Il collegamento A si realizza con un filo.

molle 13, 14, 15 e 16. Osservando lo schema possiamo vedere che le quattro resistenze da 820 Ω limitano la corrente circolante sui LED; sono interne al circuito DG11, che contiene i quattro LED, e non sono visibili.

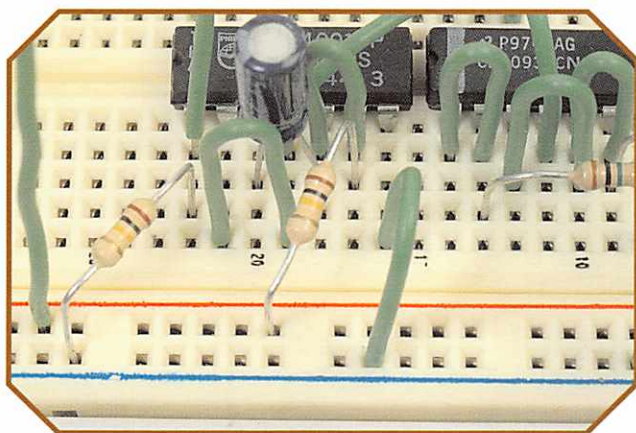
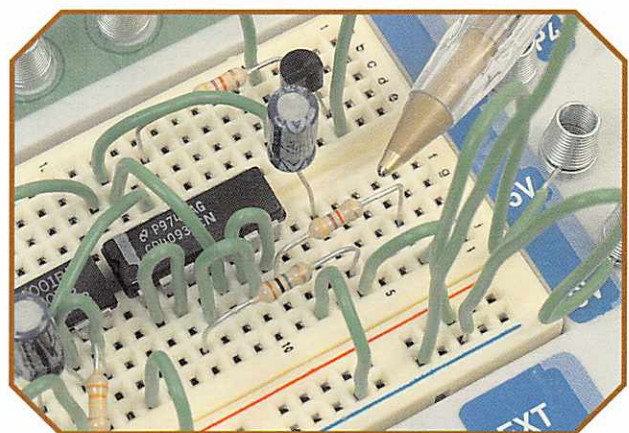
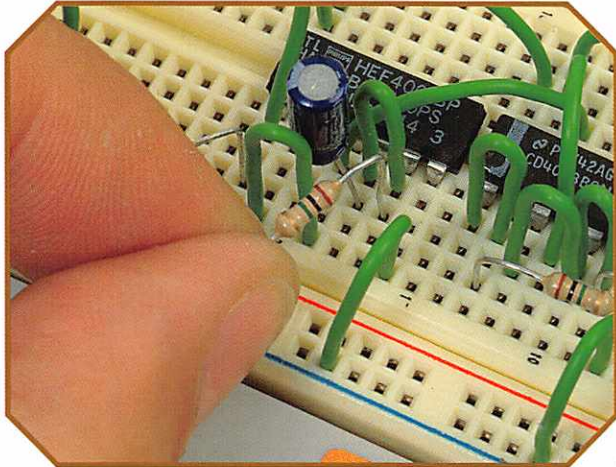
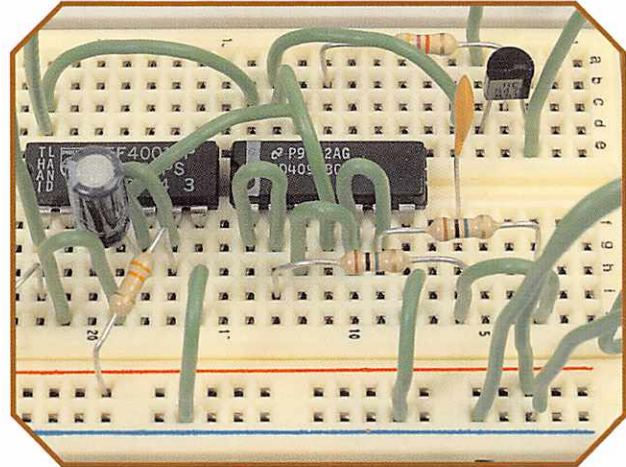
Alimentazione

Il circuito si alimenta tramite le due molle di collegamento 5 V e 0 V. I collegamenti dell'alimentazione dei LED sono precablati all'interno.

Prova

Dopo aver verificato che il circuito è montato in modo corretto, si sposta il commutatore di alimentazione sulla posizione BAT, momento in cui i quattro LED si devono illuminare. Collegando per un attimo A i LED si spengono e si accendono in modo intermittente per un certo periodo di tempo, dopodiché rimarranno illuminati in modo fisso.

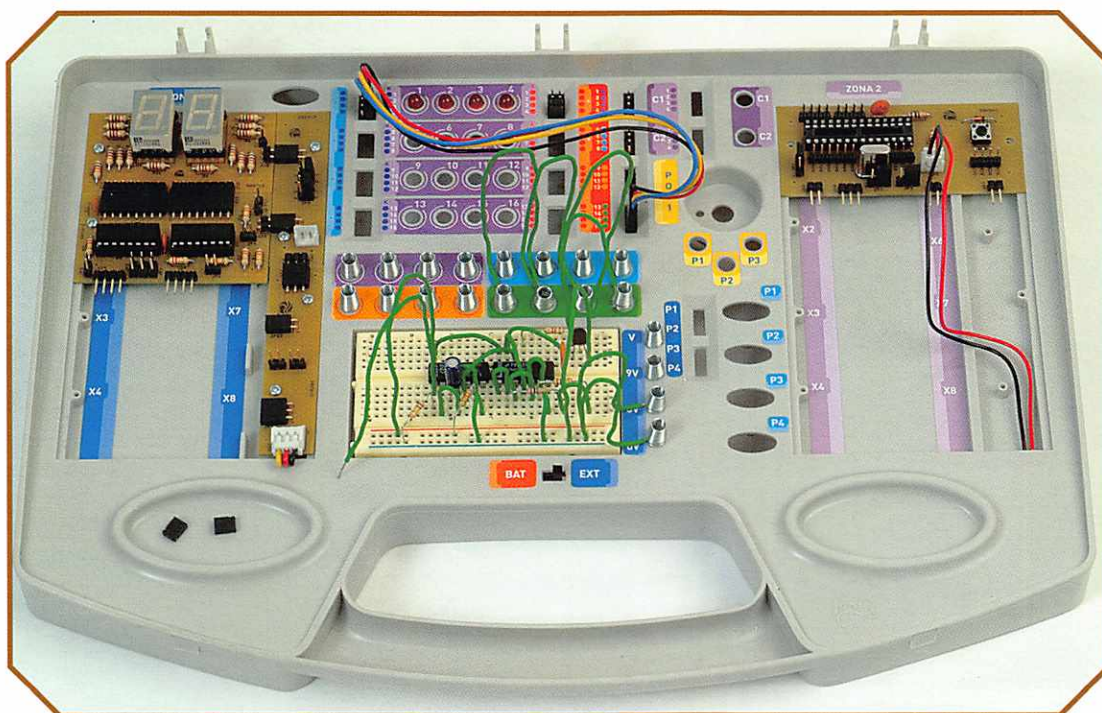
LISTA DEI COMPONENTI

U1	Circuito integrato 4001
U2	Circuito integrato 4093
Q1	Transistor BC547 o BC548
R5	Resistenza 100 K (marrone, nero, giallo)
R6	Resistenza 330 K (arancio, arancio, giallo)
R7	Resistenza 1 M (marrone, nero, verde)
R8, R9	Resistenza 4K7 (giallo, viola, rosso)
C1, C2	Condensatore 10 µF, elettrolitico

R6 determina la temporizzazione.

R8 controlla l'intermittenza dei LED.

Con R6 da 1 M
la temporizzazione aumenta.

Con R8 da 1 M e C2 da 22 nF,
continua a funzionare ma con un'altra frequenza.

Varianti

In questo circuito è possibile cambiare il valore di alcuni componenti, ad esempio, se diminuiamo R6 la temporizzazione diminuisce, possiamo provare con 100 K e con 1 M.

La resistenza R8 determina, insieme a C2, la cadenza o frequenza dell'intermittenza, possiamo ad esempio, tornare a porre R6 da 330K, passare R8 a 1 M e utilizzare per C2 la capacità di 22 nF. Potremo osservare delle variazioni nella cadenza dell'intermittenza.

Possiamo provare anche un'altra variazione, inserendo una porta collegata come invertente tra l'uscita della porta U2D e la resistenza R9. Questo cambio inverte l'illuminazione dei LED che ora rimangono spenti in stato di riposo e si illuminano solamente in modo intermittente quando si attiva l'ingresso A.

Vista generale dell'esperimento.

Il repertorio delle istruzioni

Continuiamo la presentazione delle istruzioni del microcontroller per fare in modo che il PIC cominci a interagire con l'hardware del nostro Laboratorio Digitale. In questa sezione vedremo le istruzioni di impostazione a zero e quelle di salto, facendo pratica con degli esempi.

Istruzioni di impostazione a zero

Questo gruppo di istruzioni si chiama così perché la funzione che esse svolgono è quella di impostazione a zero o cancellazione di un registro. Sono molto utili quando vogliamo resettare un valore, come ad esempio, quando eseguiamo un conteggio di tempo e trascorso il tempo desiderato, vogliamo reinizializzare il valore del conteggio. In questo gruppo sono comprese tre istruzioni:

- CLRF f: (Clear f) Il contenuto del registro f si cancella, impostiamo a zero tutti i bit del registro. Il bit Z del Registro di Stato passa automaticamente a 1.

- CLRW: (Clear W) Si cancella il contenuto del registro di lavoro W. Dato che questo registro è utilizzato in tutti i programmi è molto comodo che esista un'istruzione specifica per resettarlo. Come nell'istruzione precedente, il bit Z passa automaticamente a 1.

- CLRWDT f: (Clear Watchdog Timer) Questa istruzione resetta il valore del temporizzatore del Watchdog e il valore del suo prescaler o divisore di frequenza. Coinvolge il registro STATUS attivando (impostando a 1) i bit $\overline{TO}$ e $\overline{PD}$

Istruzioni di salto

Le istruzioni di salto rompono la normale frequenza del flusso di istruzioni del programma, provocando dei salti. Quindi queste istruzioni coinvolgono il contenuto del Contatore di Programma (PC).

In questo gruppo si possono includere le istruzioni BTFSC e BTFSS, dato che saltano un'istruzione se si compie una condizione, però abbiamo preferito inserirle nel gruppo successivo: istruzioni di gestione dei bit. Le istruzioni appartenenti al gruppo di salto sono:

- CALL k: (Call Subroutine) Chiamata a una subroutine. Prima si memorizza sullo stack l'indirizzo di ritorno, quello dell'istruzione successiva a questa (PC+1). Gli 11 bit dell'indirizzo di destinazione 'k' passano al PC <10:0>. I bit più significativi del PC si caricano mediante PCLATH. Il tempo di esecuzione di questa istruzione è di due cicli di istruzioni.

- RETLW k: (Return with Literal in W) Si carica il registro di lavoro W con il valore numerico k. Il PC acquisisce il valore dello stack (l'indirizzo di ritorno). Occupa due cicli di istruzione.

- RETFIE: (Return from Interrupt) Istruzione per ritornare da un interrupt. Si carica il PC con l'indirizzo della cima dello stack e si imposta a 1 il bit GIE di abilitazione globale degli interrupt.

- RETURN: (Return from Subroutine) Istruzione di due cicli di durata la cui funzione è quella di ritornare da una subroutine.

MNEMONICO E OPERANDI	DESCRIZIONE	CICLI	CODICE OP	FLAG
CLRF f	Resetta il valore del registro f	1	00 0001 1fff ffff	Z
CLRW	Resetta il valore del registro W	1	00 0001 0xxx xxxx	Z
CLRWDT	Resetta il valore del Watchdog Timer e del suo divisore di frequenza	1	00 0000 0110 0100	$\overline{TO}$, $\overline{PD}$

Istruzioni di impostazione a zero.

MNEMONICO E OPERANDI	DESCRIZIONE	CICLI	CODICE OP	FLAG
CALL k	Salto alla subroutine	2	10 0kkk kkkk kkkk	--
RETLW k	Ritorno da una subroutine e registro W caricato con il valore numerico K	2	11 01xx kkkk kkkk	--
RETFIE	Ritorno da un interrupt	2	00 0000 0000 1001	--
RETURN	Ritorno da una subroutine	2	00 0000 0000 1000	--
GOTO k	Salto incondizionale all'indirizzo indicato da k	2	10 1kkk kkkk kkkk	--
DECFSZ f,d	Decrementa il valore di f. Se il risultato è 0 salta l'istruzione successiva	1 o 2	00 1011 dfff ffff	--
INCFZ f,d	Incrementa il valore di f. Se il risultato è 0 salta l'istruzione successiva	1 o 2	00 1111 dfff ffff	--

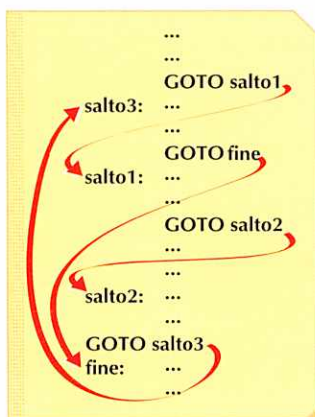
Istruzioni di salto.

- GOTO k: Salto incondizionale. Il valore immediato di 11 bit si carica sul PC (bit <10:0>). I bit più significativi del PC si caricano tramite il registro PCLATH<4:3>. Questa istruzione risulta molto utile ed è assai utilizzata in assembler, però è un'istruzione che nel resto dei linguaggi di programmazione non si utilizza o se ne limita l'uso.

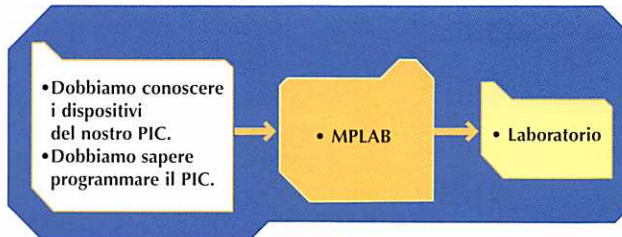
Quando eseguiamo un salto incondizionale rendiamo più complessa la struttura del programma, dato che saltiamo a un altro gruppo di istruzioni indicato mediante l'etichetta definita tramite il valore letterale, e lasciamo da eseguire le istruzioni successive a questa, a meno di non ritornare successivamente mediante un altro salto all'esecuzione sequenziale del programma. Questa istruzione non è adatta a un linguaggio di programmazione strutturato.

- DECFSZ f,d: Il contenuto del registro 'f' si riduce. In base al valore 'd' il risultato verrà scritto su W (d = 0) o sul registro 'f' (d = 1). Se il risultato è uguale a 0, si salta l'istruzione successiva e si continua con l'esecuzione.

- INCFSZ f,d: Il contenuto del registro 'f' si incrementa. Come nell'istruzione precedente, il risultato è scritto su W o sul registro 'f' in base al valore di 'd'. Se il risultato è uguale a 0, si salta l'istruzione successiva e si continua con l'esecuzione.

Programma con diversi salti incondizionali, il programma perde la sua struttura.

Calcolare il valore che contiene il registro VALORE dopo aver eseguito il seguente programma:

MOVLW	02h
MOVWF	VALORE
DECF	VALORE,1
DECFSZ	VALORE,1
INCF	VALORE,1
INCF	VALORE,1
SOLUZIONE:	VALORE = 0000 0001

Esempio di base dell'utilizzo delle istruzioni.

In un progetto realizzeremo il programma, la sua simulazione e il montaggio finale del circuito.

```
c:\progra~1\mplab\progetti\salti.asm

        LIST    P=16F870
        INCLUDE "P16F870.INC"

        ORG     0
        goto    INIZIO   ;Indirizzo di reset

INIZIO: goto    SALTO1   ;Salto all'etichetta in maiuscolo, inutile perché
                         ;il PC punterebbe direttamente a quella posizione

SALTO1: goto    salto1   ;Salto all'etichetta in minuscolo

salto1: goto    salto1   ;Ciclo infinito

Salto1: goto    Salto1   ;Non si arriva mai a questa etichetta

        END
```

Programma per fare pratica con i salti incondizionali.

Esercizi con MPLAB

Normalmente le nozioni teoriche sono affiancate da esempi. In questo modo, oltre a mettere in pratica le nuove conoscenze acquisite, è possibile imparare a utilizzare il software MPLAB con maggiore disinvoltura. È consigliabile modificare gli esempi e giocare con i programmi, come anche provare le diverse opzioni del software di simulazione. Inoltre, poco a poco, inizieremo a interagire con l'hardware del laboratorio, fino a quando saremo in grado di realizzare senza difficoltà tutte le fasi di sviluppo di un progetto.

Il primo esempio con cui ci eserciteremo è una dimostrazione di come funziona l'istruzione GOTO. Eseguite MPLAB e create un progetto con il nome, ad esempio, "salti.pjt". Ora create una nuova finestra di scrittura, scrivete il programma riportato nella figura e salvatelo con il nome "salti.asm". Associate il codice al progetto Project → Edit Project → Add Node. Fatto questo, compilate e assemblate il programma, eseguendolo passo a passo per studiarne il funzionamento. Nella figura si può vedere come rimarrà la videata di MPLAB al momento di simulare. In questo esempio il simulatore è sensibile all'utilizzo di maiuscole o minuscole, però questo si può cambiare nelle opzioni del programma.

Il programma successivo con cui ci eserciteremo è un programma dedicato a riempire e gestire un insieme di indirizzi di memoria. Utilizzando l'indirizzamento indiretto (movwf INDF) si utilizza l'indirizzo contenuto sul registro FSR per puntare il dato da muovere. Si tratta di scrivere il valore 0x18 nelle 15 posizioni di dati contigue, iniziando dall'indirizzo 0x20.

Simulazione del programma precedente dei salti con GOTO.

```
c:\progra~1\mplab\progetti\memoria1.asm

            List     P=16F870       ;Tipo di processore
            include  "P16F870.INC"  ;Definizione dei registri interni

Contatore   equ      0x0c           ;Contatore interno (c(hex)=12(decimale))
Prima       equ      0x20           ;Posizione iniziale

            org      0x00           ;Vector di reset
            goto     Inizio

            org      0x05

Inizio      movlw    .15
            movwf    Contatore      ;Carica il contatore con il valore 15 in decimale
            movlw    Prima
            movwf    FSR            ;Inizializza il puntatore con l'indirizzo iniziale
            movlw    0x18           ;Carica il valore da scrivere

Ciclo       movwf    INDF           ;Scrive il valore all'indirizzo indicato da FSR
            incf     FSR,F          ;Incrementa il puntatore FSR
            decfsz   Contatore,F    ;Decrementa il contatore sino ad arrivare a 0
            goto     Ciclo          ;Esegue l'istruzione se il contatore non è a 0

Stop        nop
            nop

            end                     ;Fine del programma sorgente
```

Esempio di gestione di memoria.

Normalmente nelle istruzioni incrementa/decrementa e salta; la struttura da seguire è quella mostrata in questo esempio: l'istruzione di incremento/decremento seguita da una di salto incondizionale. Potete vedere come mediante l'utilizzo di semplici istruzioni è possibile creare strutture complesse di controllo e cicli condizionali. Anche se il suo utilizzo non è consigliato in nessun linguaggio di programmazione, l'istruzione goto è necessaria nel nostro caso per poter realizzare strutture di controllo.

Nella figura sono riportate le finestre che abbiamo aperto per realizzare la simulazione.

Osservate come si riempiono le posizioni di memoria con il valore desiderato e al riempimento di quelle richieste, l'istruzione decfsz agisce ed evita di entrare nuovamente in ciclo, saltando l'istruzione goto.

La procedura da seguire è la stessa dell'esempio precedente. Potete nominare il progetto "memoria1.pjt" e il codice in assembler con lo stesso nome ma con l'estensione ".asm". Nella figura è riportato il codice che risolve il problema.

Come vedrete, in questo programma gestiamo le istruzioni di salto goto e decfsz.

Videata di MPLAB.

È possibile modificare il programma per vedere cosa succede. Le modifiche si devono realizzare una a una per osservare meglio il comportamento del programma.

Di seguito vi diamo alcuni suggerimenti:

– definite le variabili nelle zone di memoria occupate, ad esempio Contatore in 0x0c;

– cambiate i dati nel programma;

– ampliate il programma e fate pratica con tutte le modifiche che servono, cercando di inserire al massimo le istruzioni che abbiamo viste finora.